

Event-Equalized Dense Video Captioning

Supplementary Material

1. More implementation details

We present more details on models and the dataset. We also provide **codes of our key components** in the supplementary material.

1.1. Model details.

In this part, we introduce more details on the model structure.

Captioning Head. For each query q_i , deformable soft attention [2] is utilized to generate context features $z_{i,t}$ with frame features around the reference points. At each timestamp t , we feed context features $z_{i,t}$, event query features q_i and previous words $\{w_{i,j}\}_{j=1}^{t-1}$ into the LSTM model to get the current word $w_{i,t}$. As the sentence is generated, the captioning head produces the complete sentence $S_i = w_{i,1}, \dots, w_{i,T}$, where T denotes the length of the sentence. The process of the captioning head Cap can be formulated as:

$$w_{i,t} = Cap(z_{i,t-1}, q_i, \{w_{i,j}\}_{j=1}^{t-1}). \quad (1)$$

Event-Enhanced Encoder. This module aims to help the model focus on frame-frame and frame-event relationships. To realize this, a trainable dictionary is built with a size of $N_c * hidden_dim$. During the training process, the model updates the dictionary to separate the frames more thoroughly. For a given pseudo-event label l_i , we get a label embedding le_i from the dictionary. After that, to predict multi-scale events, the encoder adds L temporal convolutional layers (stride=2, kernel size=3) to get feature sequences across multiple resolutions, from T to $T/2^L$. Meanwhile, the pseudo-event labels are also down-scaled with 2D convolutional layers to get multi-scale labels. Finally, the multi-scale features and multi-scale pseudo-event labels are fed into the transformer encoder together.

Pseudo-Event Initialization. This module aims to help the model pay equal attention to all possible events. To realize this, center temporal location t_i is calculated for each pseudo-event E_i . After that, we calculate the Inverse Sigmoid value for t_i . The function can be formulated as :

$$InvSig(t_i) = \log\left(\frac{t_i}{1 - t_i}\right). \quad (2)$$

Then, Sinusoidal Positional Encoding is utilized to get the corresponding positional embedding. Finally, a linear projection layer and a normalization layer is used to match the hidden dimension of the decoder.

Table 1. **Ablation on the designed components.** We report the results on ActivityNet Captions. PEI denotes the Pesudo-Event initialization module. EEE denotes the Event-Enhanced encoder module. Without PEI, the queries will be randomly initialized and are updated during the training process. Without EEE, we directly fuse visual features and positional embedding for each video frame together.

PEI	EEE	BLEU4↑	METEOR↑	CIDEr↑	SODA_c↑	F1↑
×	×	2.21	8.06	29.97	5.92	54.78
×	✓	2.23	8.49	31.28	6.03	55.28
✓	×	2.36	8.40	32.41	6.09	55.57
✓	✓	2.43	8.57	33.63	6.13	56.14

1.2. Dataset Details.

In this paper, we conducted our experiments on ActivityNet Captions and YouCook2 datasets. For the ActivityNet Captions dataset, we utilize C3D, TSN, and CLIP features. The C3D features are provided by Wang et al. [2]. The TSN features are provided by Zhou et al. [3]. The CLIP features are provided by Kim et al. [1]. Among them, C3D and CLIP features are acquired by first sample video frames at a rate of 1 FPS and then extracted by the corresponding pretrained encoder. The TSN features are sampled at a rate of 2 FPS. For the YouCook2 dataset, we only use TSN and CLIP features because C3D features are not publicly available.

2. More experiments

2.1. More Ablation Experiments.

Ablation Experiments on Different Components in E²DVC on Anet Dataset. Since the components ablation study in the main paper is only conducted on the YouCook2 dataset, we present the results on the ActivityNet Captions dataset in this section. The result is presented in Table 1. It's clear that both PEI and EEE improve the model's performance on Dense Video Captioning and Event Localization tasks. The results align with that in the main paper. Combining PEI and EEE yields the best performance, demonstrating the superiority of our proposed components.

Ablation Experiments on Cluster Algorithms. As illustrated in Table 2, we present an ablation study to explore the effects of different clustering algorithms in the Event Perception Module. Here, we utilize two additional clustering algorithms (DBSCAN and KMeans) to replace the agglomeration hierarchical clustering method. The results show that three different clustering algorithms achieve significant

Table 2. **Ablation on varying cluster algorithms.** This experiment is conducted on the ActivityNet Captions and YouCook2 datasets. No_Cluster denotes the baseline without any clustering algorithm. For the dense video captioning performance, we present both captioning and localization results. We also calculate the Accuracy (Acc), Recall, and F1 score to evaluate the clustering quality. It’s obvious that the cluster quality is positive to the dense video captioning performance.

Cluster_Alg	Dataset	Dense Video Caption					Cluster Metrics		
		BLEU4↑	METEOR↑	CIDEr↑	SODA_c↑	F1↑	Acc↑	Recall↑	F1↑
No_cluster	Anet	2.21	8.06	29.97	5.92	54.78	-	-	-
DBSCAN	Anet	2.21	8.48	32.10	6.08	55.34	73.21	33.01	45.50
Kmeans	Anet	2.39	8.51	32.28	6.05	55.65	75.92	34.04	47.00
AHC	Anet	2.43	8.57	33.63	6.13	56.14	75.62	36.43	49.17
No_cluster	Anet	1.40	5.56	29.69	4.92	26.81	-	-	-
DBSCAN	YC2	1.54	5.84	30.77	5.11	27.07	41.45	28.78	33.94
Kmeans	YC2	1.57	6.00	31.76	5.21	28.08	42.31	29.04	34.44
AHC	YC2	1.68	6.11	34.26	5.39	28.64	42.54	29.76	35.02

Table 3. **Ablation on a varying number of clusters.** We report the results on the ActivityNet Captions dataset. The **best** and **second** performance results are highlighted.

N_c	BLEU4↑	METEOR↑	CIDEr↑	SODA_c↑	F1↑
0	2.21	8.06	29.97	5.92	54.78
1	2.15	8.25	30.89	6.02	54.97
2	2.25	8.30	31.78	<u>6.18</u>	55.00
3	2.27	8.46	31.58	6.01	55.24
4	<u>2.34</u>	<u>8.49</u>	32.92	6.08	55.63
5	2.43	8.57	33.63	6.13	56.14
6	2.24	8.39	<u>33.34</u>	6.35	<u>55.79</u>
7	2.19	8.38	32.41	6.02	55.43

improvements compared to the baseline without any clustering algorithm. It shows that our E²DVC can be well adapted to different clustering algorithms, which proves the effectiveness and robustness of our method. Among the three clustering algorithms, the hierarchical agglomerative clustering method outperforms DBSCAN and KMeans. This can probably be attributed to the nature of hierarchical agglomerative clustering, which does not assume any particular shape or sizes of the cluster. Sometimes some video frames do not belong to any an arbitrary event, they should be regarded as noisy points or outliers. Although DBSCAN is designed to identify and handle outliers, it’s tricky to tune the ideal parameters (eps and minPts). Meanwhile, KMeans is best suited for spherical clusters and struggle to handle irregular or noisy clusters well. These constraints may limit the flexibility of DBSCAN and KMeans.

Parameter Analysis on the Number of Clusters N_c on

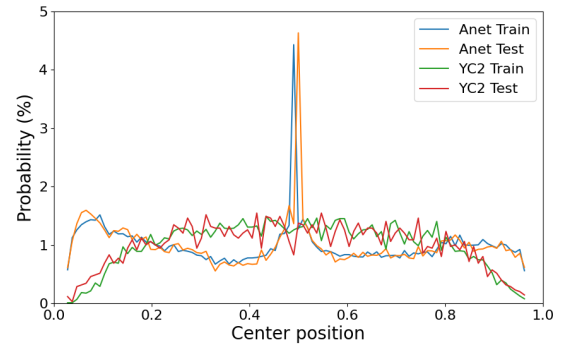


Figure 1. **Distribution of Event center position.** We present the statistics of the probability of different event center positions from ActivityNet Captions and YouCook2 datasets. The horizontal and vertical axes represent the normalized center position and event probability. It’s obvious that the center timestamps of the video has the highest Event Probability in all these datasets.

Anet Dataset. In this section, we try different numbers of clusters N_c on the ActivityNet Captions dataset. We present the results in Table 3. The Event Threshold τ is set to 4. The first row is the result from the baseline. As can be seen from the Table, when N_c is set to 1, the performance is a little higher than the baseline. This is because in this case all video frames are clustered together and the model will add an additional query to focus on the middle of the video. Luckily, as shown in Figure 1, the middle of the video has the highest probability of an event happening. As the cluster number increases, video frames will be clustered into more categories based on visual differences. The best performance appears when N_c is set to 5. If N_c continues to grow, the performance drops because the model’s atten-

Table 4. **Ablation on varying temporal event threshold τ .** We report the results on the ActivityNet Captions dataset. The **best** and the **second** performance results are highlighted.

τ	BLEU4 \uparrow	METEOR \uparrow	CIDEr \uparrow	SODA_c \uparrow	F1 \uparrow
-	2.21	8.06	29.97	5.92	54.78
0	2.18	8.35	32.58	6.25	55.05
1	2.29	8.59	31.26	6.01	53.47
2	<u>2.43</u>	8.64	32.43	6.00	54.19
3	2.49	<u>8.62</u>	34.07	6.29	55.53
4	<u>2.43</u>	8.57	<u>33.63</u>	6.13	<u>56.14</u>
5	<u>2.43</u>	8.48	32.82	<u>6.26</u>	55.88
6	2.31	8.41	32.06	6.24	56.16
7	2.29	8.40	31.75	6.15	55.37

Table 5. **Ablation on decoder queries.** The experiments are conducted on the ActivityNet Captions Dataset. The "Baseline" is the result of our baseline with the number of queries set to 10. "Random" is the result of setting the number of queries the same as our PEI module but all queries are randomly initialized. "PEI" is the result with only the PEI module implemented. "E²DVC" is the result from our model.

Method	BLEU4 \uparrow	METEOR \uparrow	CIDEr \uparrow	SODA_c \uparrow	F1 \uparrow
Baseline	2.21	8.06	29.97	5.92	54.78
Random	2.22	8.29	29.24	5.48	54.96
PEI	2.36	8.40	32.41	6.09	55.57
E ² DVC	2.43	8.57	33.63	6.13	56.14

tion is distracted and the real events may be neglected. This aligns with the experiments on the YouCook2 dataset in the main paper.

Parameter Analysis on the Event Threshold τ on Anet Dataset. This parameter decides the lower bound of pseudo-events duration. It's utilized to detect and discard isolated frames which could be outliers. The results are shown in Table 4. The first row presents the result of our baseline. All performance is better than the baseline in this table. However, when τ is too small, the performance is not better than the baseline by a large margin. That's because even isolated frames are allocated certain decoder queries to focus on. This will distract the model's attention and make the real events being overlooked. The best performance is achieved when τ is set to 2-5. When τ continues to grow, the performance will decrease because more and more possible events are discarded. Finally, the result will be degraded to be the same as the baseline. **Ablation Experiments on Decoder Queries.** In this part, we conducted an ablation experiment on the decoder queries. The number

Table 6. **Evaluation on uneven samples.** The experiments are conducted on the ActivityNet Captions Dataset. We separate all events into four different groups with their durations and compare our method with the baseline on all these groups.

Method	Duration	B4	M	C	F1
Base	0-10s	1.44	7.74	30.54	52.14
Base	10-20s	1.67	7.85	30.91	54.31
Base	20-30s	1.87	8.00	31.09	55.34
Base	>30s	2.24	8.34	32.63	56.34
E ² DVC	0-10s	1.66 \pm 15.3%	8.12 \pm 4.9%	32.85 \pm 7.6%	53.91 \pm 3.4%
E ² DVC	10-20s	1.75 \pm 4.7%	8.17 \pm 4.1%	32.92 \pm 6.6%	55.41 \pm 2.0%
E ² DVC	20-30s	2.08 \pm 11.2%	8.19 \pm 2.4%	33.11 \pm 6.5%	56.31 \pm 1.7%
E ² DVC	>30s	2.55 \pm 13.8%	8.45 \pm 1.3%	33.67 \pm 3.2%	56.43 \pm 0.1%

of decoder queries is set to 10 on ActivityNet Captions in our baseline. However, in EPM, this decoder query number is decided by the number of pseudo-events. We observe that when N_c is set to 5 and τ is set to 4, the average number of decoder queries equals to 16, which is bigger than 10. Therefore, to investigate whether the performance improvement is brought by the increased parameter count from adding decoder queries or genuinely driven by our proposed Pseudo-Event Initialization, we conducted comparative experiments by setting the query number to 16 in the baseline. The experimental results are shown in Table 5. From the table, we can observe that simply adding the number of decoder queries ("Random") is worse than our PEI module. Actually, the result from "Random" is only comparable with the "Baseline" but with increased computational cost. This proves that the performance improvement in our method is not only brought by the increased weight number. Our visual clustering method can provide convincing pseudo-events to the decoder and help the model focus on important locations.

2.2. Evaluation on uneven samples.

In this section, we split all events into four groups according to their durations and compare our method with the baseline on these groups to further validate the effectiveness of the model. We present the results on the ActivityNet Captions Dataset in Table 6. From the table, we can make two observations: 1) Short events are more difficult to caption than long ones. 2) The smaller the duration of events, the greater performance increase is achieved by our model. This demonstrates that by assigning equal attention to all events, our model will not overlook short ones, thereby achieving better localization and captioning for short events and thus improving overall performance.

2.3. More Qualitative Results.

In this section, we visualize two more examples on the ActivityNet Captions dataset. The results are shown in Figure 2. As can be seen from the Figure, our result aligns with the ground truth with high localization accuracy and

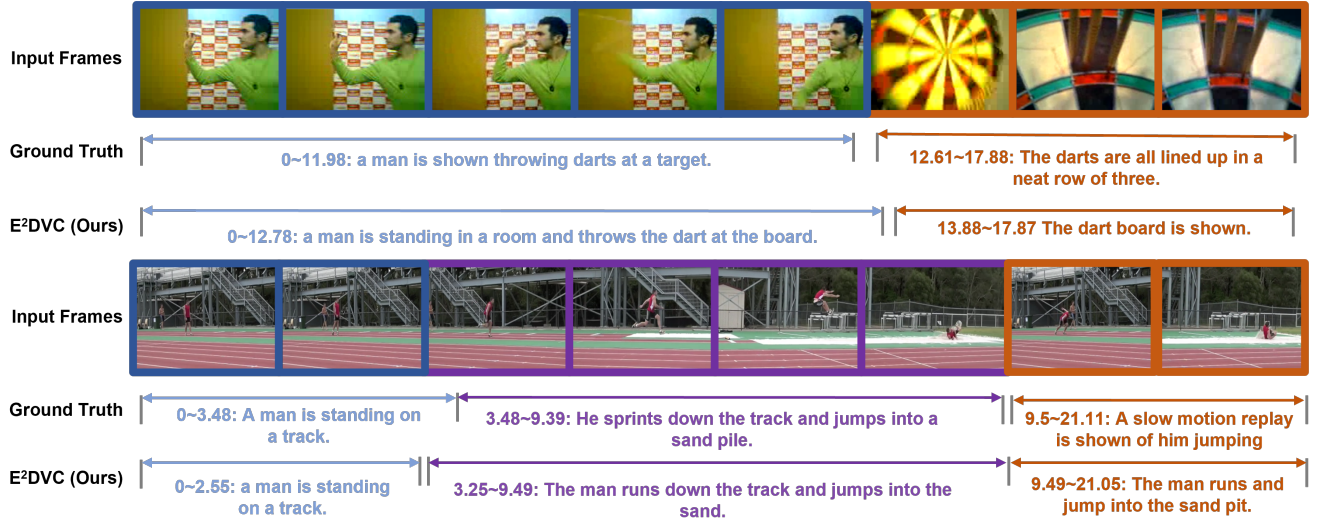


Figure 2. **Example visualizations of dense event captioning prediction.** The color of the image border represents the category of the pseudo-event. From top to down, we show the results from the ground truth and our method.

captioning performance.

References

- [1] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? dense video captioning with cross-modal memory retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13904, 2024. 1
- [2] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6847–6857, 2021. 1
- [3] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018. 1